Systèmes multi-agents

Rapport du projet

Auteur
GHOUIBI Ghassen

Encadrée par
Mr Ali Cherif

Table des matières

I Introduction

Arduino est une plate-forme électronique open source basée sur du matériel et des logiciels faciles à utiliser.
Les cartes Arduino sont conçus pour lire des informations en entrées en plusieurs formes capteurs, un button poussoir, une lumière ... etc ensuite transformer cet entré en sortie par exemple pour allumer une LED, activer un moteur etc ...
En effet on peut indiquer à notre carte quoi faire en envoyant tout simplement quelque insctructions au microcontrôleur, pour parvenir à çelà on utilise le langage de programmation Arduino trés proche de C/C++ IDE Arduino sachant qu'on peut programmer sur d'autres language comme Python par exemple.

Au fil des années, Arduino a été le cerveau de milliers de projets, des objets du quotidien aux instruments scientifiques complexes.
Une communauté mondiale de créateurs, étudiants, amateurs, artistes, programmeurs et professionnels s'est réunie autour de cette plate-forme open source, leurs contributions ont ajouté une quantité incroyable de connaissances accessibles qui peuvent être d'une grande aide pour les novices et les experts.

Toutes les cartes Arduino sont entièrement open-source, permettant aux utilisateurs de les construire indépendamment et éventuellement de les adapter à leurs besoins particuliers. Le logiciel est également open-source, et il se développe grâce aux contributions des utilisateurs du monde entier.

I .1 Objectif

Réaliser grâce un Arduino Uno un circuit qui nous permettera de lire la température à traves un module WIFI sur notre téléphone.
La réalisation de cette tâche nous demandera tous d'abord de choisir nos composants dans notre cas ça sera :

- Arduino Uno
- Module DS1621 (température)
- Module DS1621 (Wifi)
- Câble
- Breadboard arduino
- Résistances & Led

On va utiliser aussi dans notre projet l'outils `Fritzing` qui nous permet de visualiser plus prêt nos circuit et surtout les schémas qui seront dans ce rapport.

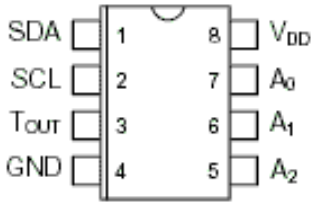
II Modules

II .1 Arduino Uno

Arduino, est une marque de cartes électroniques matériellement libres sur lesquelles se trouve un microcontrôleur (d'architecture Atmel AVR comme l'Atmega328p, et d'architecture ARM).
Le microcontrôleur peut être programmé pour analyser et produire des signaux électriques, de manière à effectuer des tâches très diverses comme la domotique, le pilotage d'un robot, de l'informatique embarquée, etc.

II .2 Module Température DS1621

Le DS1621 est un thermomètre digital en I2C. Ce module va nous servir pour relever la température en $^\circ$C à l'aide d'un Arduino.
Voici le brochage de ce module :



II .3 Module Wifi ESP8266

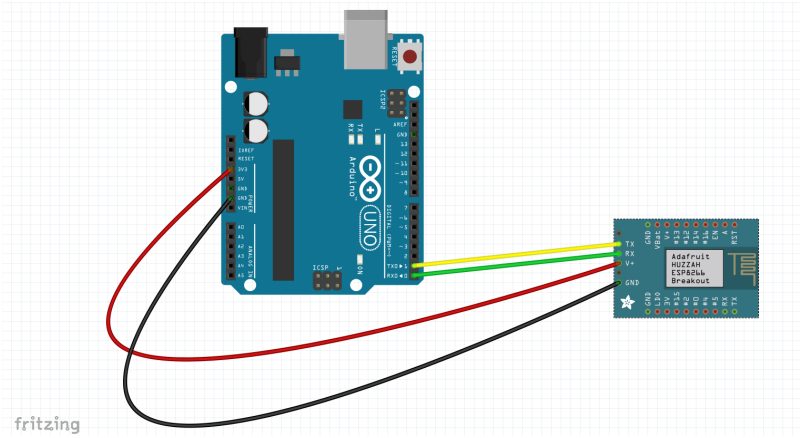
L'ESP8266 est un circuit intégré à microcontrôleur avec connexion Wi-Fi développé par plusieurs entreprise et il existe plusieurs variante de ce dernier.
Le version ESP-01 est trés connu pour la taille réduite et qui permet de connecter un microcontrôleur à un réseau Wi-Fi et d'établir des connexions TCP/IP avec des commandes Hayes.

III Branchement

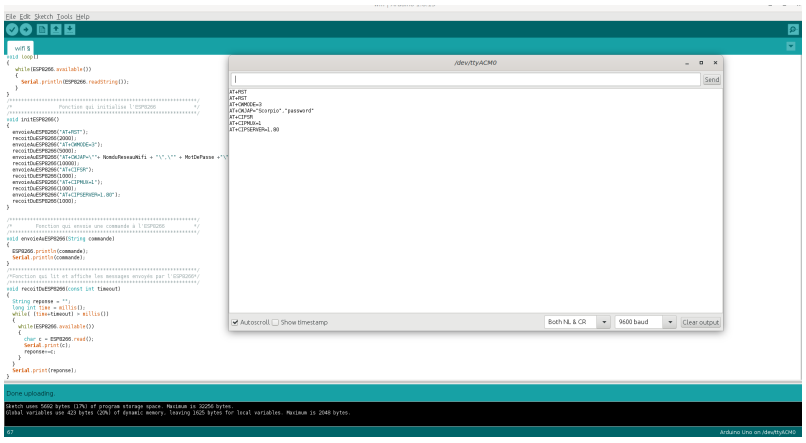
Dans cette section on expliquer le branchement de nos modules sur notre arduino pour çelà on va utiliser un logiciel qui s'appelle `Fritzing`[1].

1. Fritzing est un logiciel libre de conception de circuit imprimé qui permet de concevoir de façon entièrement graphique le circuit et d'en imprimer le typon. (Wikipedia)

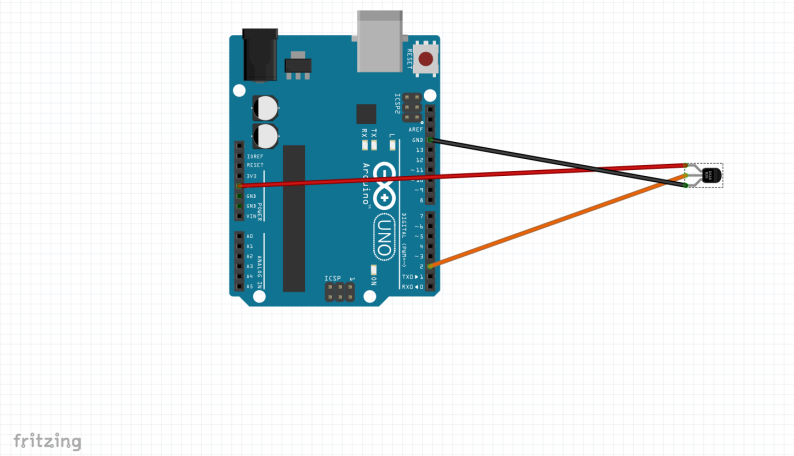
III .1 Module WIFI



En théorie l'ESP s'alimente 3.3V donc il faudrait adapter les niveaux des tensions sur la liaison série mais elle possède un circuit de protection jusqu'a 5.8V donc on peut s'en passer.
Ensuite il faut lancer l'interface de développement arduino et ouvrir le moniteur série, configurer la liaison série en `Both NL & CR` à 9600 Baud.
Maintenant on va tout simplement utiliser des commandes AT pour interagir avec le ESP, et configurer le ESP pour qu'il puisse agir en tant que serveur TCP.
Finalement aprés la connexion sur notre WIFI avec la commande `AT + CWJAP` en précisant le nom du WIFI et le mot de passe, pour savoir nos IP il suffit de taper la commande `AT + CIFSR` maintenant on peut se connecter sur notre serveur grâce à notre IP soit avec une application ou même un petit programme pour l'échange de données dans notre cas ça sera relever la température.



III .2 Module température



Dans l'exemple ci-dessous on n'utilise pas le même capteur vu qu'il n'était pas disponible sur Fritzing c'est une variante de capteur de température.

IV Code

IV .1 Module WIFI

```
#include <SoftwareSerial.h>
SoftwareSerial ESP8266(10, 11);
String NomduReseauWifi = "Scorpio";
String MotDePasse      = "password";
void setup(){
    Serial.begin(9600);
    ESP8266.begin(9600);
    initESP8266();
}
void loop(){
    while(ESP8266.available()){
    Serial.println(ESP8266.readString());
    }
}
void initESP8266(){
envoieAuESP8266("AT+RST");
recoitDuESP8266(2000);
envoieAuESP8266("AT+CWMODE=3");
recoitDuESP8266(5000);
envoieAuESP8266("AT+CWJAP=\""+ NomduReseauWifi +"\",\"" + MotDePasse +"\"");
recoitDuESP8266(10000);
envoieAuESP8266("AT+CIFSR");
recoitDuESP8266(1000);
envoieAuESP8266("AT+CIPMUX=1");
```

```
25                  recoitDuESP8266(1000);
26                  envoieAuESP8266("AT+CIPSERVER=1,80");
27                  recoitDuESP8266(1000);
28                  }
29                  void envoieAuESP8266(String commande){
30                      ESP8266.println(commande);
31                  }
32
33                  void recoitDuESP8266(const int timeout){
34                      String reponse = "";
35                      long int time = millis();
36                      while( (time+timeout) > millis()){
37                          while(ESP8266.available()){
38                              char c = ESP8266.read();
39                              reponse+=c;
40                          }
41                      }
42                  Serial.print(reponse);
43                  }
44
```

Les commandes AT utiliser dans ce code sont :

— AT + RST : pour
— AT + CWMODE : mode Client & serveur
— AT + CWJAP : nom et mot de passe de wifi
— AT + CIFSR : récuprer les adresse IP
— AT + CIPMUX : commencer un serveur avec l'ESP
— AT + CIPSERVER : commencer le serveur sur le port 80 en HTTP

IV .2 Module Température DS1621

```
1  #include <Wire.h>
2
3  #define DEV_ID 0x90 >> 1
4
5  void setup() {
6  Serial.begin (9600);
7
8  Wire.begin ();
9  Wire.beginTransmission (DEV_ID);
10 Wire.write (0xAC);
11 Wire.write (0x02);
12 Wire.beginTransmission (DEV_ID);
13 Wire.write (0xEE);
14 Wire.endTransmission ();
15 }
16
```

```
17
18 void loop() {
19 unsigned int data;
20
21 Wire.beginTransmission (DEV_ID);
22 Wire.write (0xAA);
23 Wire.endTransmission ();
24 Wire.requestFrom (DEV_ID, 2);
25 if (Wire.available ()) data  = Wire.read () << 8;
26 if (Wire.available ()) data |= Wire.read ();
27
28 Serial.print (data >> 8);
29 Serial.print (data & 0xFF ? ".5" : ".0");
30 Serial.println("");
31
32 delay (1000);
33 }
34
```

V Résultat

Finalement, on voulait utiliser `removeXY`[2] pour connecter notre adruino à notre téléphone malheureusement le module WIFI avait un problème aprés avoir essayer les solutions suivantes :

- Inversion des fils Rx et Tx
- Retrait du microcontrôleur de la platine Arduino
- Ajustement du baud-rate à 115200

2. `https://remotexy.com/`